Notice of Allowability	Application No.	Applicant(s)
	09/964,237	BATES ET AL.
	Examiner	Art Unit
	J. Derek Rutten	2192

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address--

All claims being allowable, PROSECUTION ON THE MERITS IS (OR REMAINS) CLOSED in this application. If not included herewith (or previously mailed), a Notice of Allowance (PTOL-85) or other appropriate communication will be mailed in due course. **THIS NOTICE OF ALLOWABILITY IS NOT A GRANT OF PATENT RIGHTS.** This application is subject to withdrawal from issue at the initiative of the Office or upon petition by the applicant. See 37 CFR 1.313 and MPEP 1308.

1. ☒ This communication is responsive to *Request for reconsideration filed 12/02/2005*.

2. ☒ The allowed claim(s) is/are *1-3,5-7 and 9-25*.

3. ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

    a) ☐ All   b) ☐ Some*   c) ☐ None of the:

      1. ☐ Certified copies of the priority documents have been received.

      2. ☐ Certified copies of the priority documents have been received in Application No. _____ .

      3. ☐ Copies of the certified copies of the priority documents have been received in this national stage application from the International Bureau (PCT Rule 17.2(a)).

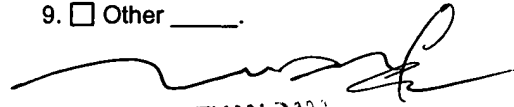   * Certified copies not received: _____ .

Applicant has THREE MONTHS FROM THE "MAILING DATE" of this communication to file a reply complying with the requirements noted below. Failure to timely comply will result in ABANDONMENT of this application. **THIS THREE-MONTH PERIOD IS NOT EXTENDABLE.**

4. ☐ A SUBSTITUTE OATH OR DECLARATION must be submitted. Note the attached EXAMINER'S AMENDMENT or NOTICE OF INFORMAL PATENT APPLICATION (PTO-152) which gives reason(s) why the oath or declaration is deficient.

5. ☐ CORRECTED DRAWINGS (as "replacement sheets") must be submitted.

    (a) ☐ including changes required by the Notice of Draftsperson's Patent Drawing Review (PTO-948) attached

      1) ☐ hereto or 2) ☐ to Paper No./Mail Date _____ .

    (b) ☐ including changes required by the attached Examiner's Amendment / Comment or in the Office action of Paper No./Mail Date _____ .

    Identifying indicia such as the application number (see 37 CFR 1.84(c)) should be written on the drawings in the front (not the back) of each sheet. Replacement sheet(s) should be labeled as such in the header according to 37 CFR 1.121(d).

6. ☐ DEPOSIT OF and/or INFORMATION about the deposit of BIOLOGICAL MATERIAL must be submitted. Note the attached Examiner's comment regarding REQUIREMENT FOR THE DEPOSIT OF BIOLOGICAL MATERIAL.

Attachment(s)

1. ☒ Notice of References Cited (PTO-892)

2. ☐ Notice of Draftperson's Patent Drawing Review (PTO-948)

3. ☐ Information Disclosure Statements (PTO-1449 or PTO/SB/08), Paper No./Mail Date _____

4. ☐ Examiner's Comment Regarding Requirement for Deposit of Biological Material

5. ☐ Notice of Informal Patent Application (PTO-152)

6. ☒ Interview Summary (PTO-413), Paper No./Mail Date _____ .

7. ☒ Examiner's Amendment/Comment

8. ☒ Examiner's Statement of Reasons for Allowance

9. ☐ Other _____ .

TUAN DAM
SUPERVISORY PATENT EXAMINER

DETAILED ACTION

1.   This action is responsive to Applicant's response dated 12/02/2005, responding to the

10/19/2005 Office action provided in the rejection of claims 1-25, wherein no claims have been

amended, no claims have been canceled, and no new claims have been added.

2.   Applicant's request for reconsideration of the finality of the rejection of the last Office

action is persuasive and, therefore, the finality of that action is withdrawn.

EXAMINER'S AMENDMENT

3.   An examiner's amendment to the record appears below. Should the changes and/or

additions be unacceptable to applicant, an amendment may be filed as provided by 37 CFR

1.312. To ensure consideration of such an amendment, it MUST be submitted no later than the

payment of the issue fee.

Authorization for this examiner's amendment was given in a telephone interview with

Gero G. McClellan, Reg. No. 44227, on 12/15/2005. The Examiner adopted the claim

amendments proposed by Mr. McClellan as presented after an Examiner-initiated interview on

December 15, 2005 (See attached "Examiner-Initiated Interview Summary").

The application has been amended as follows:

Please cancel claims 4 and 8, and amend claims 1, 2, 5-7, and 14-20. A complete listing

of the claims with the proper status identifiers appears below.

LISTING OF THE CLAIMS:

1.   (Currently Amended) A method of debugging an application in a debugging environment comprising the application and a debugger program, the method comprising:

automatically counting a number of times each breakpoint in the application is encountered while the application is executing during a test run, wherein the number is increasing and, at any time during the test run, always reflects a current number of times a given breakpoint has been encountered during the test run and wherein counting the number is not limited by a predetermined number representative of a desired number of encounters of the given breakpoint; and

storing the number for each breakpoint in a memory space for use in a subsequent run; wherein automatically counting comprises, for each breakpoint:

(i) incrementing a breakpoint-specific counter each time a breakpoint associated with the breakpoint-specific counter is encountered in a particular code segment; and

(ii) resetting the breakpoint-specific counter each time the particular code segment is entered.

2.   (Currently Amended) The method of claim 1, further comprising, while the application is stopped, receiving a user-input request to uninterruptedly execute the application until a user-specified breakpoint is encountered some number of times, N-X, where N is a stored number of times the user-specified breakpoint was encountered during the test run and X is a value equal to or greater than zero and less than N.

3.   (Original)   The method of claim 1, further comprising, after the application is stopped at a location in response to a last breakpoint encounter of a particular breakpoint encountered Y number of times at the last breakpoint encounter, receiving a user-input request to uninterruptedly execute the application until the application is again stopped at the location in response to encountering the particular breakpoint Y number of times.

4.   (Canceled)

5.   (Currently Amended) A <u>tangible</u> computer readable <u>storage</u> medium containing a debug program which, when executed by a computer configured with an application being debugged during a debug session, performs breakpoint counter operations, the debug program comprising:

counting instructions for automatically counting a number of times each breakpoint in the application is encountered, wherein the number is increasing and, at any time, always reflects a current number of times a given breakpoint has been encountered and wherein counting the number is not limited by a predetermined number representative of a desired number of encounters of the given breakpoint; and

storing instructions for storing the number for each breakpoint in a memory space for use in a subsequent run<u>;</u>

<u>wherein the counting instructions, when executed:</u>

<u>(i) increment a breakpoint-specific counter each time a breakpoint associated with the breakpoint-specific counter is encountered in a particular code segment; and</u>

<u>(ii) reset the breakpoint-specific counter each time the particular code segment is entered</u>.

6.   (Currently Amended) The <u>tangible</u> computer readable <u>storage</u> medium of claim 5, wherein the debug program further comprises, execution instructions for uninterruptedly executing the application until a user-specified breakpoint is encountered N-X times, wherein the execution instructions are configured for execution in response to a user request specifying the user-specified breakpoint and a number of encounters N-X, where N is a stored number of times the particular breakpoint was encountered during the test run and X is a value equal to or greater than zero <u>and less than N</u>.

7.   (Currently Amended) The <u>tangible</u> computer readable <u>storage</u> medium of claim 5, wherein the debug program further comprises, execution instructions for, after the application is

stopped at a last stopped location of the test run in response to encountering a last encountered breakpoint Y number of times, processing a user-input request to uninterruptedly execute the application until the application is again stopped at the location in response to again encountering the last encountered breakpoint Y number of times.

8.   (Canceled)

9.   (Previously Presented)   A method of debugging an application in a debugging environment comprising the application and a debugger program, the method comprising:

associating a counter with a breakpoint and with at least one application code segment in which the breakpoint is located;

incrementing the counter each time the breakpoint is encountered, wherein the counter, at any time, always reflects a current number of times a given breakpoint has been encountered and wherein incrementing the counter is not limited by a predetermined number representative of a desired number of encounters of the given breakpoint; and

resetting the counter each time the application code segment is entered.

10.   (Original)   The method of claim 9, further comprising storing a counter value of the counter for use in a subsequent execution of the application.

11.   (Original)   The method of claim 9, wherein resetting the counter comprises firing an internal breakpoint which does not call a user interface.

12.   (Original)   The method of claim 9, further comprising:

determining whether a counter value of the counter has reached a user-specified value; and if so, halting execution of the application and issuing a user notification indicative of the counter value.

13.   (Original)   The method of claim 9, further comprising:

uninterruptedly executing the application until a counter value of the counter has reached a user-specified value; and then halting execution of the application.

14.   (Currently Amended) A tangible computer readable storage medium containing a debug program which, when executed by a computer configured with an application being debugged during a debug session, performs breakpoint counter operations, the debug program comprising counter instructions which, when executed:

        associate a counter with a breakpoint and with at least one application code segment in which the breakpoint is located;

        increment the counter each time the breakpoint is encountered, wherein the counter, at any time, always reflects a current number of times a given breakpoint has been encountered and wherein incrementing the counter is not limited by a predetermined number representative of a desired number of encounters of the given breakpoint; and

        reset the counter each time the application code segment is entered.

15.   (Currently Amended) The tangible computer readable storage medium of claim 14, wherein the code segment is one of a routine and a loop.

16.   (Currently Amended) The tangible computer readable storage medium of claim 14, wherein the debug program further comprises execution instructions for uninterruptedly executing the application until a counter value of the counter is equal to a user-specified value.

17.   (Currently Amended) The tangible computer readable storage medium of claim 14, wherein the debug program further comprises, storing instructions for storing a counter value of the counter for use in a subsequent execution of the application.

18.   (Currently Amended) The tangible computer readable storage medium of claim 14, wherein the debug program further comprises internal breakpoint setting instructions for setting an internal breakpoint configured to reset the counter without calling a user interface when the internal breakpoint is fired.

19.   (Currently Amended) The tangible computer readable storage medium of claim 14, wherein the debug program further comprises counter value instructions for determining whether a counter value of the counter has reached a user-specified value and, if so, halting execution of the application and issuing a user notification indicative of the counter value.

20.   (Currently Amended) The tangible computer readable storage medium of claim 14, wherein the debug program further comprises counter value instructions for allowing uninterrupted execution of the application until a counter value of the counter has reached a user-specified value and then halting execution of the application.

21.   (Previously Presented)   A computer system, comprising:
        a memory containing content comprising at least a debug program to implement a debug session, a breakpoint table configurable with breakpoint-specific counters and an application for debugging; and a processor which, when executing at least a portion of the content during the debug session, is configured to:
                associate a breakpoint-specific counter with a breakpoint and with at least one application code segment in which the breakpoint is located;
                increment the counter each time the breakpoint is encountered, wherein the counter, at any time, always reflects a current number of times a given breakpoint has been encountered and wherein incrementing the counter is not limited by a predetermined number representative of a desired number of encounters of the given breakpoint; and
                reset the counter each time the application code segment is entered.

22.   (Original)   The computer system of claim 21, wherein the debug program, when executed by the processor, configures the processor to store a counter value of the breakpoint-specific counter in the breakpoint table for use in an execution of the application.

23.   (Original)   The computer system of claim 21, wherein the debug program, when executed by the processor, configures the processor to set an internal breakpoint configured to

reset the breakpoint-specific counter without calling a user interface when the internal breakpoint is fired.

24.   (Original)   The computer system of claim 21, wherein the debug program, when executed by the processor, configures the processor to uninterruptedly execute the application until a counter value of the breakpoint-specific counter is equal to a user-specified value.

25.   (Original)   The computer system of claim 22, wherein the debug program, when executed by the processor, configures the processor to issue a user notification indicative of the counter value.

--End Examiner's Amendment--

4.   The following is an examiner's statement of reasons for allowance:

The examiner indicated that this application would be in condition for allowance if independent claims 1 and 5 are amended to include the features of claims 4 and 8 which include incrementing a breakpoint counter each time a breakpoint in a section of code is encountered, and resetting the breakpoint counter each time the section of code is entered. U.S. Patent 6,751,751 to Murray et al. discloses a breakpoint counter configured to trigger a breakpoint event after a certain number of breakpoint conditions are encountered (column 3 line 51 – column 4 line 4). However, this breakpoint counter is not configured to be reset each time a particular code segment is entered. The above features, taken in combination with all remaining features of the independent claim are not taught or suggested by the prior art of record. The applicant agreed to amend the independent claims 1 and 5 as indicated by the examiner. Independent

claims 9, 14, and 21 also recite the above features. The distinctions provided by the independent

claims apply equally to all dependent claims. Thus all pending claims 1-3, 5-7, and 9-25 are

allowed.

Any comments considered necessary by applicant must be submitted no later than the

payment of the issue fee and, to avoid processing delays, should preferably accompany the issue

fee. Such submissions should be clearly labeled "Comments on Statement of Reasons for

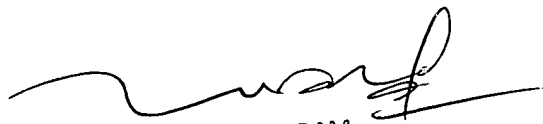Allowance."

Conclusion

Any inquiry concerning this communication or earlier communications from the

examiner should be directed to J. Derek Rutten whose telephone number is (571) 272-3703. The

examiner can normally be reached on T-F 6:00 - 4:30.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's

supervisor, Tuan Q. Dam can be reached on (571) 272-3695. The fax phone number for the

organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent

Application Information Retrieval (PAIR) system. Status information for published applications

may be obtained from either Private PAIR or Public PAIR. Status information for unpublished

applications is available through Private PAIR only. For more information about the PAIR

system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR

system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

jdr

TUAN DAM
SUPERVISORY PATENT EXAMINER